Overview 7

This chapter describes the THINK Project Manager and how to write a program using the THINK Project Manager. The rest of the chapters in this part of the manual discuss specific aspects of the THINK Project Manager.

Contents

The THINK Project Manager 117

The Project 117

Writing a Program 117
- Creating source files. 118
- Adding libraries 118
- Compiling the program. 118
- Running the program 118
- Debugging the program 119
- Building the application 119

Using the THINK Project Manager 119

7 Overview

The THINK Project Manager

The THINK Project Manager is a completely integrated development environment, not just a compiler for the Macintosh. Traditional development environments consist of three separate applications: the editor, the compiler, and the linker. It is up to you to create your source files with a text editor, run each file through the compiler, and finally link all your object files.

In the THINK Project Manager, these components work together as parts of the same application. This way, a compiler knows when you've edited a file. The compiler produces object code that the linker can put together in an instant. Then the THINK Project Manager can launch your program. And because the THINK Project Manager is still running, it can launch the source-level debugger so you can debug your program.

The Project

For more information on the project, read Chapter 8, "Organizing Projects."

The project is at the heart of the THINK Project Manager. It contains a list of all the files that form your program. What you see on the screen is a project window, listing all the project's files. You can choose what appears next to each file, such as its object code size and data size.

Rather than producing a separate binary object code file, the THINK Project Manager keeps all object code in the project document, in ready-to-link form.

Because the project document knows all the files that make up your program (including header files and resource files), it can keep track of changes. When you edit a source file, the project manager marks that file for recompilation. When you edit a header file, the project manager marks all the files that use that file for recompilation.

Writing a Program

Writing a program in the THINK Project Manager is like writing a program in any other development environment. You create your source files, compile them, then link the object code to create an executable file. The difference is that the same application can do all of this.

Creating source files

For more information on editing your files, read Chapter 10, "Editing Files."

When you write a program with the THINK Project Manager, you create a project document. The project document is in a folder for all the files related to your program. Next create your source files and add your libraries. The source files are standard text files, so you can use existing source files. The THINK editor provides some features that help you edit source code. Its search facilities include a pattern-matching option based on Grep, and a multi-file search that looks for strings in any file in your project.

Adding libraries

Every compiler guide has a chapter on the Toolbox. Read that chapter for more information on how to call Toolbox routines with a compiler.

Virtually every program you write will need to access the Macintosh Toolbox. The THINK Project Manager comes with libraries that let you call any Macintosh Toolbox routine exactly as it's described in *Inside Macintosh*. In THINK C and Symantec C++, the code for Toolbox routines marked `[Not In ROM]` as well as the glue code needed to call some of the other Toolbox routines are in the MacTraps and MacTraps2 libraries. MacTraps handles the functions in both System 6 and System 7 (except for the Gestalt Manager). MacTraps2 handles functions added for System 7.

Every compiler guide has a section describing the libraries it comes with. Read that section for more information.

The THINK Project Manager compilers also come with other libraries that you can use in your programs. For example, THINK C and Symantec C++ come with copies of the ANSI standard library, which contains the functions described in the ANSI standard, and the Unix library, which contains Unix system functions including memory calls. Symantec C++ also comes with several class libraries, including IOStreams and Complex.

You can create your own libraries to include in your projects or distribute to other people.

Compiling the program

For more information on compiling your program, read Chapter 11, "Running Programs."

The THINK Project Manager knows when files need to be recompiled. If you edit a source file, the project manager marks it for recompilation. If you edit a header file, the project manager marks all the files that depend on that file. You can ask the THINK Project Manager to bring your project up to date, or rely on the auto-make facility to do it for you when you run the program.

Running the program

For more information on running your program, read Chapter 11, "Running Programs."

The THINK Project Manager lets you run your program from it. The project manager recompiles all marked source files and loads any

unloaded libraries. Then the linker links all your code together instantly.

The THINK Project Manager launches your program as if you had double-clicked it from the Finder. This way, you know exactly how your program will behave in actual conditions. If you're running under System 7 or MultiFinder under System 6, the THINK Project Manager launches your program in its own partition. Since the THINK Project Manager is still running, you can look at your source files while your program is running.

Debugging the program

For more information on the debugger, read Chapter 12, "Debugging Programs."

To help get your program working correctly, use the THINK Project Manager's source-level debugger. It lets you step through your code, set breakpoints, and examine and modify variables. You can set conditional breakpoints that stop execution only when certain conditions are true.

Building the application

For more information on building projects, read Chapter 13, "Building Projects."

Finally, when you're ready to put together the final application, the THINK Project Manager's smart linker examines the object code in your project to make the final file as small as possible.

Using the THINK Project Manager

The best way to learn how to use the THINK Project Manager is to follow one of the tutorials in Part One, "Getting Started." The rest of the chapters in this part of the manual describe in detail the components of the entire THINK Project Manager . The following summary shows the basic steps for writing a program with the THINK Project Manager:

1. Start the Think Project Manager. Double-click the THINK Project Manager icon from the Finder.

2. Create a new project. When the THINK Project Manager starts, it asks you (with a standard file dialog) to open an existing project or create a new one. Click the New button. Select the type of project you want to create from the

second dialog, and enter the name of the project in the third dialog.

3. Create source files. Use the **New** command in the **File** menu to get an empty edit window. To open existing source files, use the **Open** command in the **File** menu.

4. Save source files. Use the **Save** command in the **File** menu to save source files. Source files for THINK C must end in `.c`, and source files for Symantec C++ must end in `.cpp` or `.cp`.

5. Add source files. Use the **Compile** command in the **Source** menu to compile the active source file and add it to the project window automatically. If you want to add a source file without compiling it, use the **Add** command in the **Source** menu to add the frontmost source file. The name of the file is in the command name. Use the **Add Files** command to add any source files that you haven't opened with the THINK Editor.

6. Add libraries. Use the **Add Files** command in the **Source** menu to add libraries to the project window. A dialog appears with two lists: the top one lists contents of the current folder, the bottom one lists files that the THINK Project Manager will add to your project. Add files to the bottom list with the Add and Add All buttons. When you're finished, click the Done button.

7. Run project. Use the **Run** command in the **Project** menu to run your project. If there are source files that need to be compiled or libraries that need to be loaded, the THINK Project Manager asks you if you want to bring the project up to date. Click the Update button. If you want to use the source-level debugger, choose the **Use Debugger** command to turn it on before running your program.

8. Build application. Use the **Build Application** command to turn your project into a stand-alone application. A standard file dialog prompts you for the name of your application.